

BSM946 – Workshop 1: Exploring STATA and a sample dataset

TASK 1: Open STATA

This first workshop will be a little different to the rest as it's aim is to get you started on STATA. In future labs you will be doing a lot more and reading a lot less!

Basic STATA information

The window labeled *Command* is where you type your commands. STATA then shows the results in the larger window immediately above, called appropriately enough *Results*. Your command is added to a list in the window labeled *Review* on the left, so you can keep track of the commands you have used. The window labeled *Variables*, on the top right, lists the variables in your dataset. The *Properties* window immediately below that, displays properties of your variables and dataset.

STATA's interface allows you to select commands and options from a drop down menu and dialog system. However, I strongly recommend using the command language as a way to ensure reproducibility of your results. The drop down menu can be helpful when you are starting to learn STATA, particularly because after you point and click on the menus and dialogs, STATA types the corresponding command for you.

STATA Commands

STATA commands are case-sensitive, for example, `display` is not the same as `Display` or `DISPLAY` and the latter two will *not* work.

STATA comes with a few sample data files, let's start by opening one, `lifeexp.dta`, which has data on life expectancy and gross national product (GNP) per capita in 1998 for 68 countries

TASK 2: Open the sample dataset `lifeexp.dta`

```
sysuse lifeexp
```

TASK 3: Find out about the data in the file.

Try to use the command `describe`, `browse` and try to use the command `summarize` command followed by variable names

e.g. `summarize lexp gnppc`

We see that live expectancy averages 72.3 years and GNP per capita ranges

from \$370 to \$39,980 with an average of \$8,675. We also see that STATA reports only 63 observations on GNP per capita, so we must have some missing values. Missing values in STATA are coded as “.”.

TASK 4: list the countries for which we are missing GNP per capita

```
list country gnppc if missing(gnppc)
```

We see that we have indeed five missing values. This example illustrates a powerful feature of STATA: the action of any command can be restricted to a subset of the data.

If we had typed `list country gnppc` we would have listed these variables for all 68 countries. Adding the *condition* `if missing(gnppc)` restricts the list to cases where gnppc is missing.

TASK 5: Graph how life expectancy varies with GNP per capita

Try using the command `graph twoway scatter` followed by the two variable names. What does the graph show?

TASK 6: Create a new variable which is the log of the existing gnppc variable

Try using the command `gen newvariablename = log(gnppc)`

TASK 7: Browse the data to make sure the new variable has been created properly

TASK 8: Why has STATA reported that there are 5 missing values? Which observations do these belong to?

TASK 9: Let's run our first STATA regression to see how life expectancy is determined by the log(gnppc)

We want to use the following specification:

$$\text{Life Expectancy} = \beta_0 + \beta_1 \text{Log(GNPPC)} + \varepsilon$$

You need to use the `regress` command, listing the dependent variable after it and then the list of independent variables

```
regress dependentvar independentvars
```

Note that the regression is based on only 63 observations. STATA omits observations that are missing the outcome or one of the predictors. The log of GNP per capita "explains" 61% of the variation in life expectancy in these countries. We also see that a one percent increase in GNP per capita is associated with an increase of 0.0277 years in life expectancy. (To see this point note that if GNP increases by one percent its log increases by 0.01.)

TASK 10: Visualize the regression we have just run

We can then plot the data against our regression line to visualize the fit of our regression.

```
graph twoway (scatter lexp loggnppc) (lfit lexp loggnppc)
```

In this command each expression in parenthesis is a separate two-way plot to be overlayed in the same graph. The fit looks reasonably good, except for a possible outlier.

Concluding Notes:

ALWAYS USE A DO -FILE! A do-file is just a set of STATA commands typed in a plain text file. To access STATA's do editor use click Window | Do-file Editor | New Do-file Editor in the menu system.

If you save your list of commands from the *command* window into a do-file, you can then click once to execute the file. This means that you can move from your unclean data, to the clean data and regression results in one single step. If you ever make a mistake in your data cleaning you can then easily correct this, rather than if you save the dataset numerous times which saves the error into the data.

TASK 11: Create a do-file from today's lab and save it